(12) UK Patent Application (19) GB (11) 2 340 001 (13) A

(43) Date of A Publication 09.02.2000

(21) Application No 9912801.9

(22) Date of Filing 03.06.1999

(30) Priority Data
(31) 09095252 (32) 10.06.1998 (33) US

(51) INT CL[7]
G06F 17/30 // H04L 29/06

(52) UK CL (Edition R)
H4P PPEC
G4A AUDB

(56) Documents Cited
EP 0921481 A WO 97/18635 A

(58) Field of Search
UK CL (Edition Q) H4P PPEC PPG
INT CL[6] G06F 17/30 , H04L 12/56 29/06
Online Databases: WPI, EPODOC, JAPIO

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

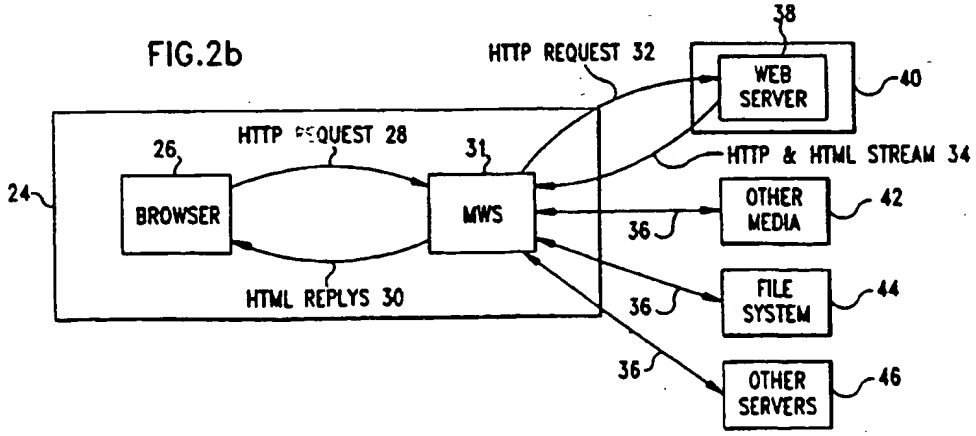
(72) Inventor(s)
Jacob Paul Ukelson

(74) Agent and/or Address for Service
S R Davies
IBM United Kingdom Limited, Intellectual
Property Dept, Hursley Park, WINCHESTER,
Hampshire, SO21 2JN, United Kingdom

(54) Abstract Title
Method for enhancing operation of a HTML web browser

(57) A client computer 24 has a standard HTML web browser 26 and a separate micro web server (MWS) 31 allowing the browser to transparently access remote servers 40 46, file systems 44 and other media 42. The MWS accepts URL requests from the browser, determines the source media (e.g. CD, DVD, the Web) and retrieves the requested data using a search engine. The MWS translates different data types to HTML, translates different protocols to HTTP either on the fly or as pages are stored on the disk. Compressed data is decompressed and additional data may be added such as advertisements and consistent buttons to all pages before transmission to the browser. The MWS also provides selective user defined access to web pages, password protection and pre-fetch of linked web pages.

FIG.2b



GB 2 340 001 A

FIG. 1

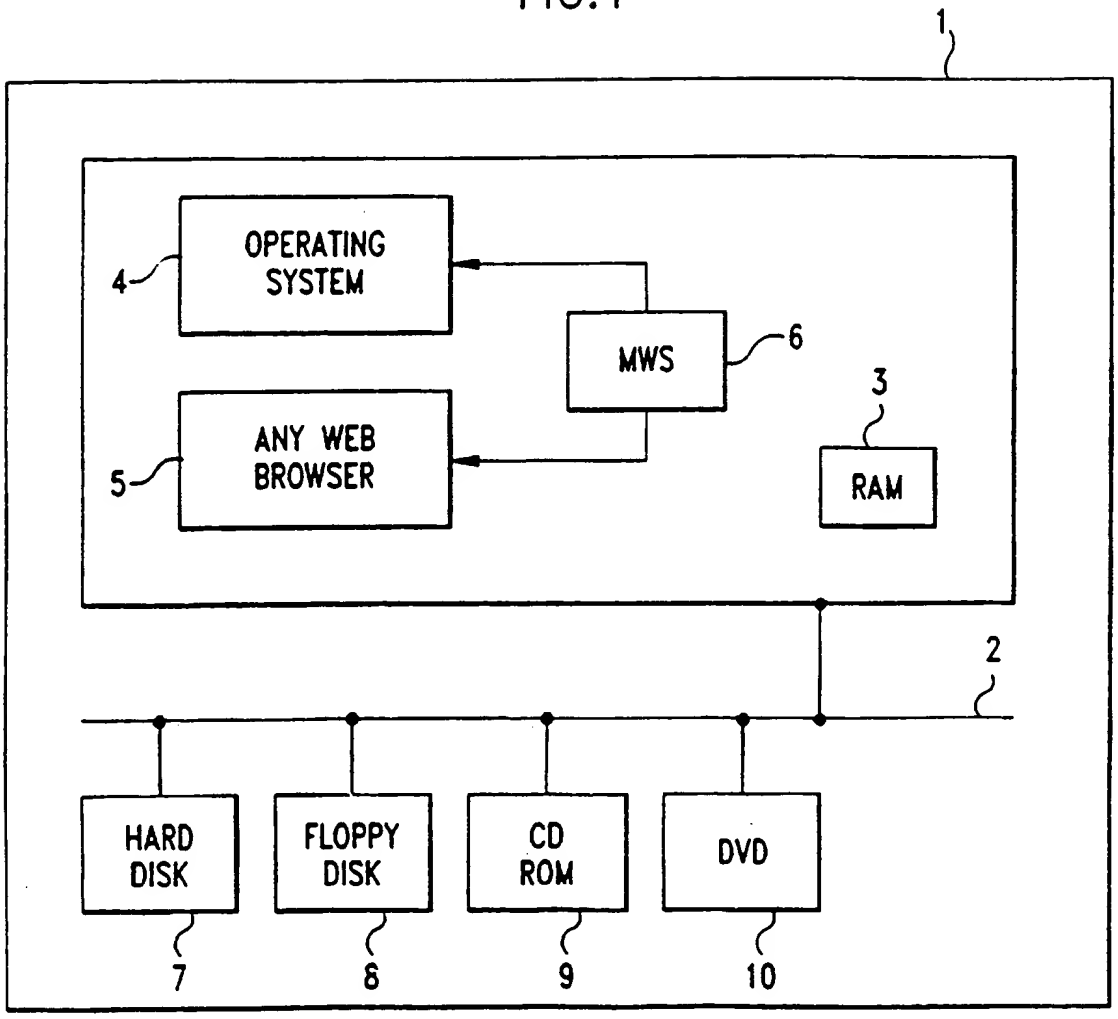

1
OPERATING SYSTEM
4
MWS
6
3
ANY WEB BROWSER
5
RAM
2
HARD DISK
FLOPPY DISK
CD ROM
DVD
7
8
9
10

FIG.2a

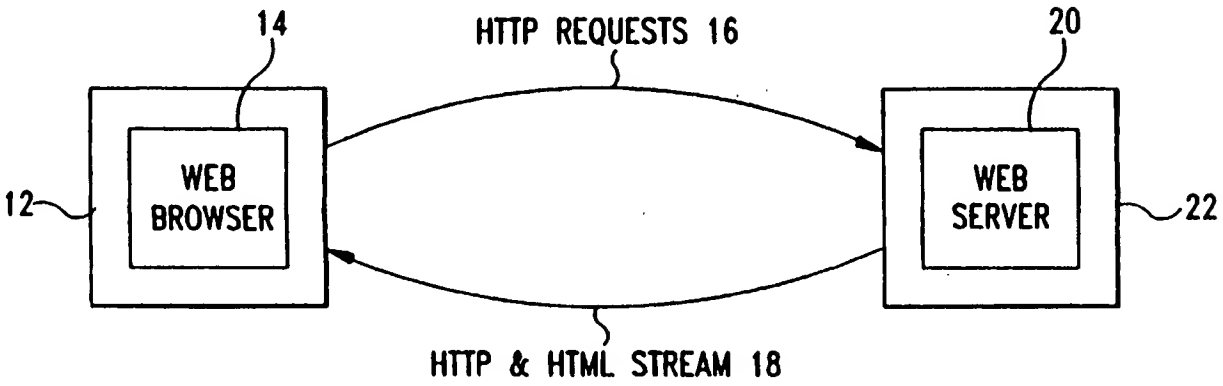
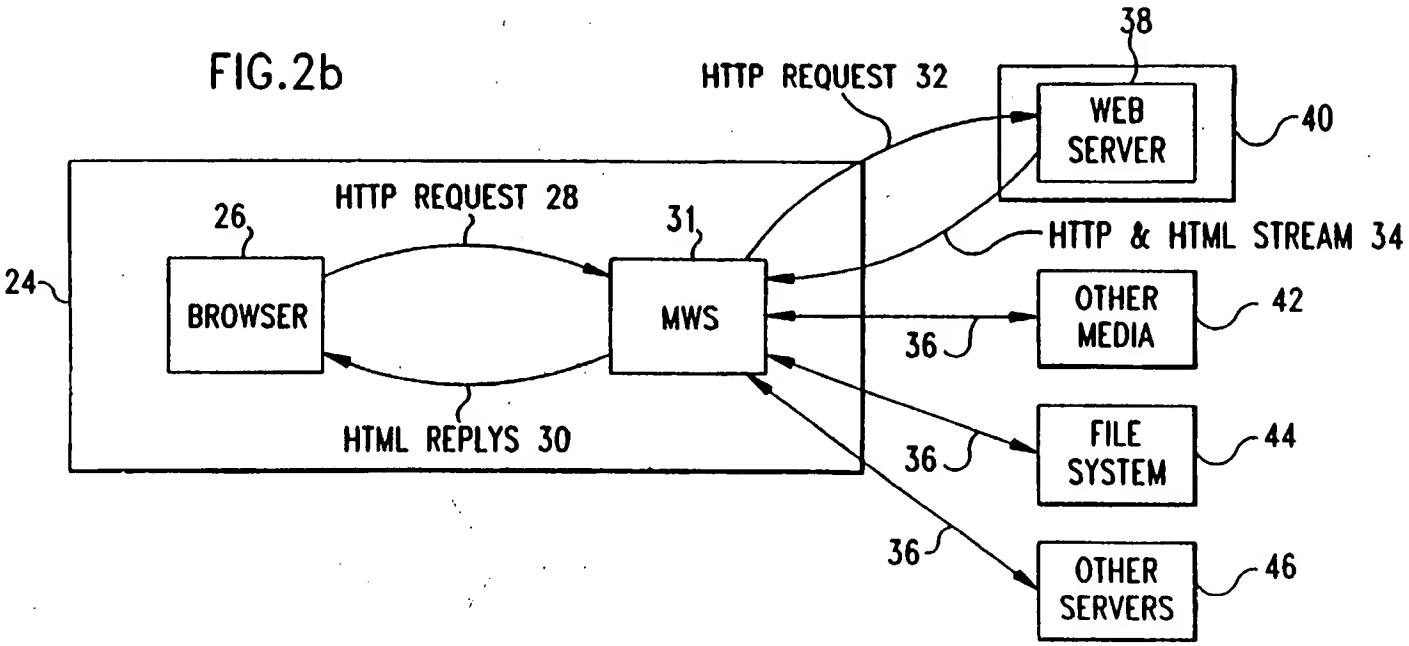


FIG.2b



217

FIG.3

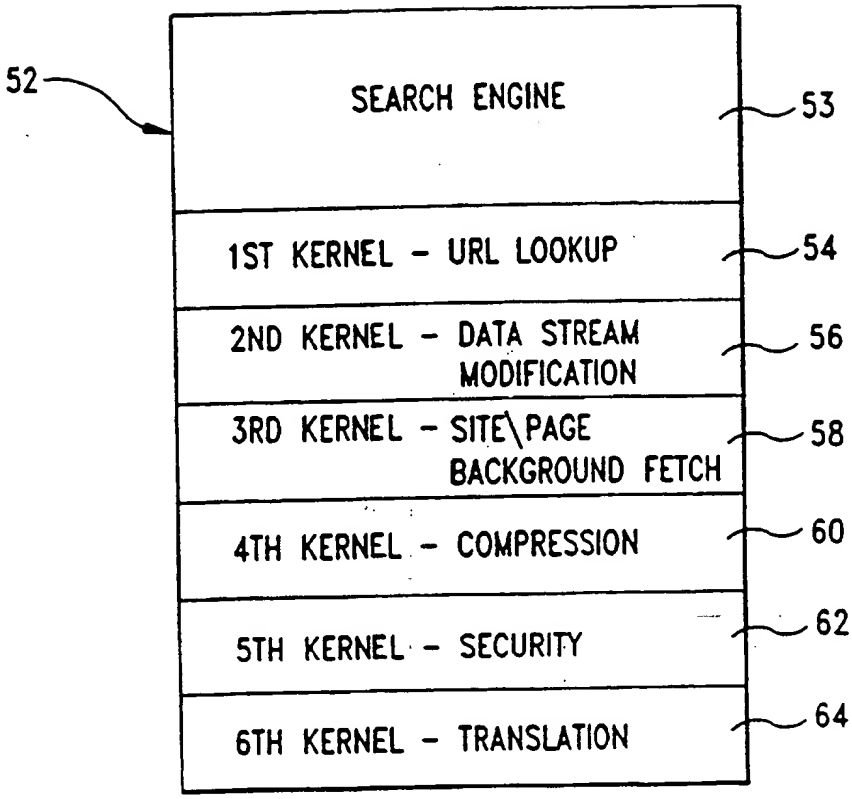
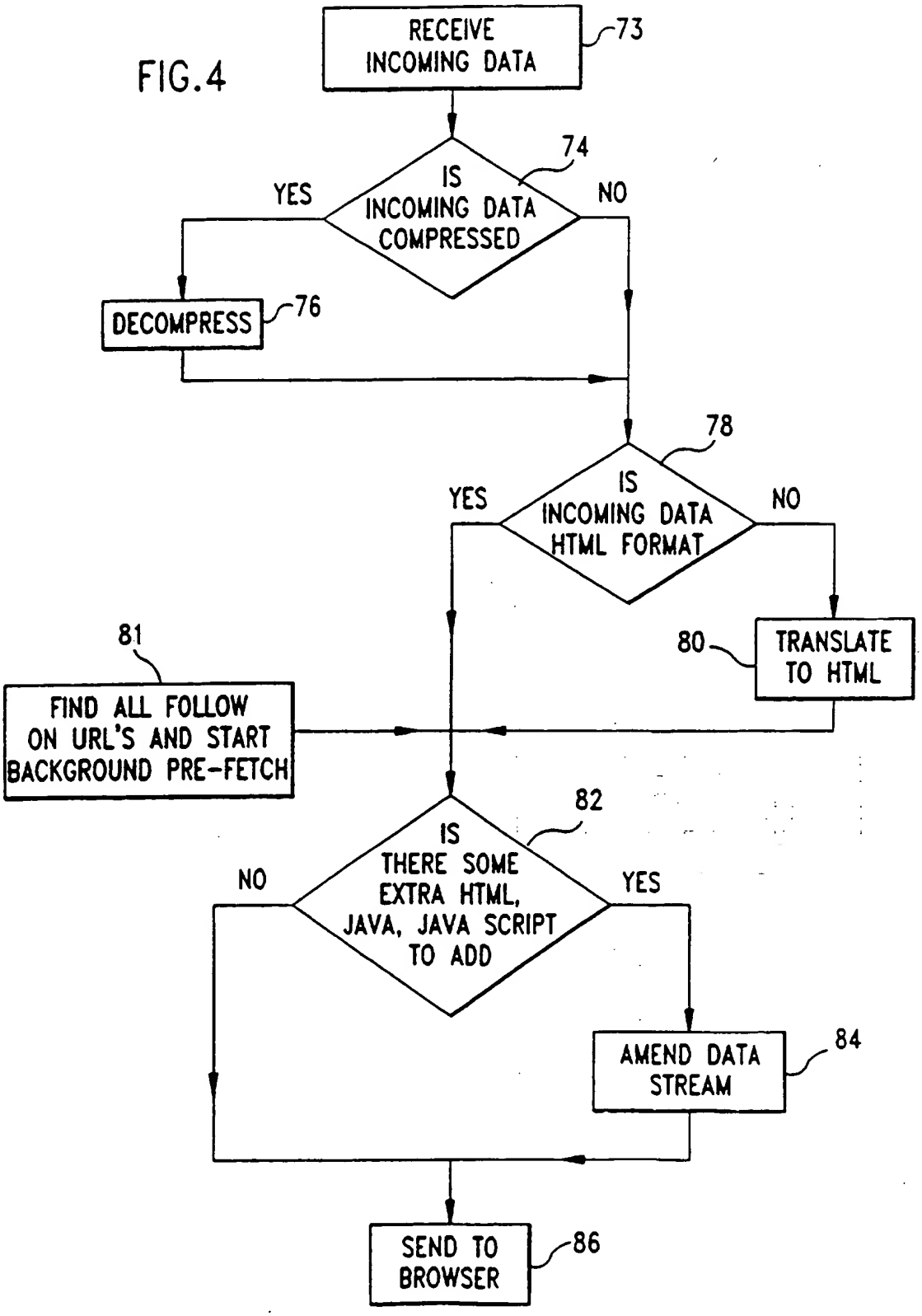

52
SEARCH ENGINE
53
1ST KERNEL - URL LOOKUP
54
2ND KERNEL - DATA STREAM MODIFICATION
56
3RD KERNEL - SITE\PAGE BACKGROUND FETCH
58
4TH KERNEL - COMPRESSION
60
5TH KERNEL - SECURITY
62
6TH KERNEL - TRANSLATION
64

FIG.4


RECEIVE INCOMING DATA
73
IS INCOMING DATA COMPRESSED
74
YES
NO
DECOMPRESS
76
IS INCOMING DATA HTML FORMAT
78
YES
NO
TRANSLATE TO HTML
80
FIND ALL FOLLOW ON URL'S AND START BACKGROUND PRE-FETCH
81
IS THERE SOME EXTRA HTML, JAVA, JAVA SCRIPT TO ADD
82
NO
YES
AMEND DATA STREAM
84
SEND TO BROWSER
86

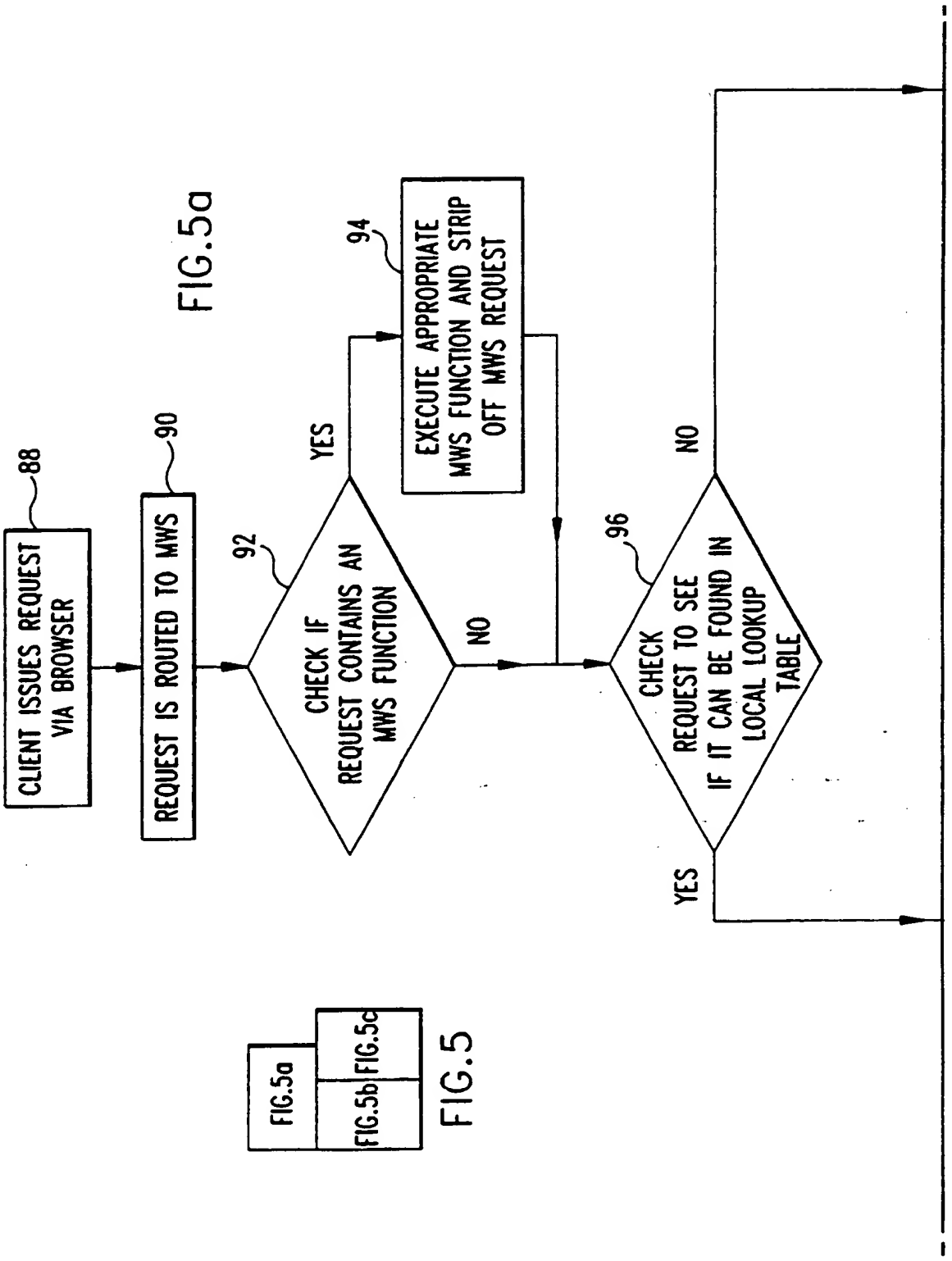

CLIENT ISSUES REQUEST VIA BROWSER
88
REQUEST IS ROUTED TO MWS
90
CHECK IF REQUEST CONTAINS AN MWS FUNCTION
92
YES
NO
EXECUTE APPROPRIATE MWS FUNCTION AND STRIP OFF MWS REQUEST
94
CHECK REQUEST TO SEE IF IT CAN BE FOUND IN LOCAL LOOKUP TABLE
96
NO
YES

FIG.5a

FIG.5a
FIG.5b
FIG.5c

FIG.5

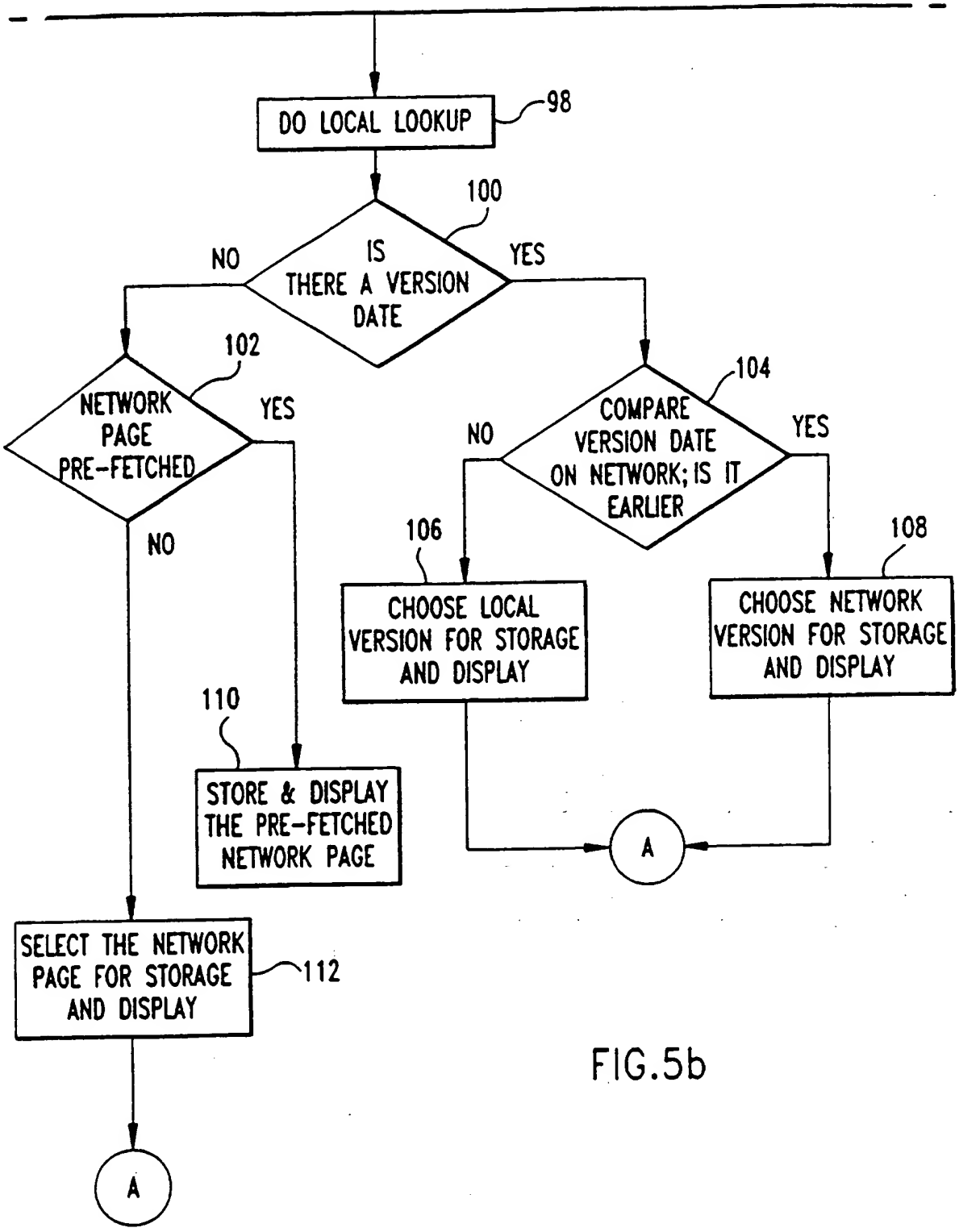

DO LOCAL LOOKUP
98
100
IS THERE A VERSION DATE
NO
YES
102
NETWORK PAGE PRE-FETCHED
YES
NO
104
COMPARE VERSION DATE ON NETWORK; IS IT EARLIER
NO
YES
106
CHOOSE LOCAL VERSION FOR STORAGE AND DISPLAY
108
CHOOSE NETWORK VERSION FOR STORAGE AND DISPLAY
110
STORE & DISPLAY THE PRE-FETCHED NETWORK PAGE
A
SELECT THE NETWORK PAGE FOR STORAGE AND DISPLAY
112
A

FIG.5b

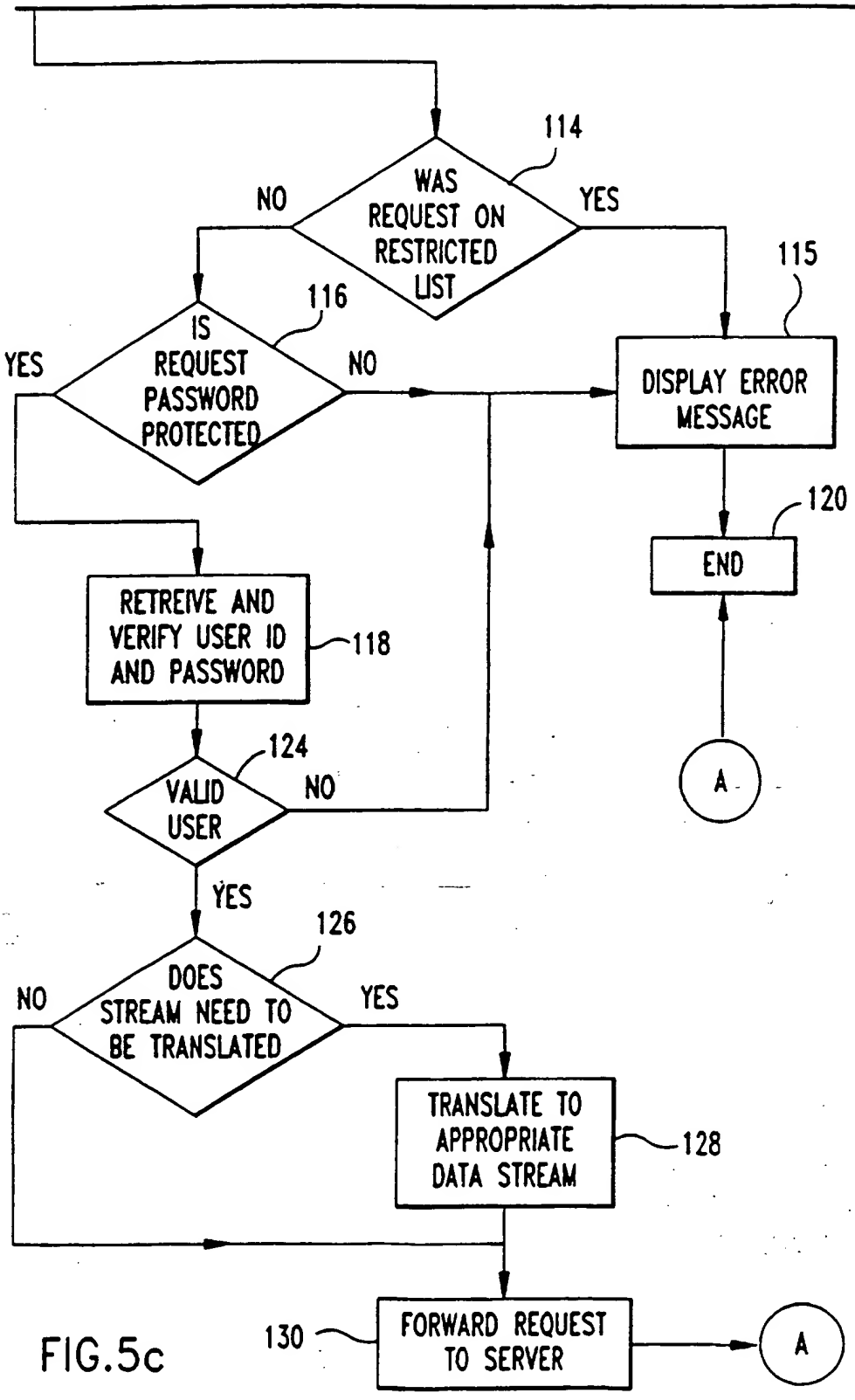

114
WAS REQUEST ON RESTRICTED LIST
NO
YES
115
116
IS REQUEST PASSWORD PROTECTED
YES
NO
DISPLAY ERROR MESSAGE
120
END
RETREIVE AND VERIFY USER ID AND PASSWORD
118
124
VALID USER
NO
A
YES
126
DOES STREAM NEED TO BE TRANSLATED
NO
YES
TRANSLATE TO APPROPRIATE DATA STREAM
128
130
FORWARD REQUEST TO SERVER
A

FIG.5c

2340001

METHOD AND APPARATUS FOR SUPPORTING OR OPERATING A WEB BROWSER

The present invention relates generally to computer networks and, more particularly to a method for enhancing the operation of a client browser.

The Internet, and more particularly the World Wide Web, continues to receive tremendous publicity. It is a collection of interconnected computer networks that covers the entire globe. The network of computers which collectively comprise this phenomenon has grown at a staggering rate, almost doubling in size every ten months for the past six years. The data available has grown as fast as the Internet itself and measures in the terabytes.

The World Wide Web is that part of the Internet which represents all the computers (servers) that offer users access to hypermedia-based information and documentation. Hypermedia enables users to navigate the Internet, moving with point-and-click ease from one location or one document to another. Browsers provide a graphical interface to the Web with menu options, icons and images you click on, buttons, graphics, and links that you use to access files of information from the Web. These files are known as "documents." A Web document can be just one page, or it can be several pages. Even if the document is just one page, you usually find links to other documents, and from those documents are links to yet others. The first document you access usually has an entry "point," or home page, so named because it usually contains the creator's name, a company name if its a business, and pointers to the document contents.

The World Wide Web uses several protocols to transport and display the multimedia resources that reside on computers (servers) around the world. One of them is the HTTP (HyperText Transport Protocol). HTTP works with Web servers to provide a client-server environment for the Internet. HTTP supports the ability of the Internet to provide access to an enormous quantity of interlinked resources.

The basic model of the Internet is straightforward, it can be thought of as a global client/server application. A client program (Web browser) needs to know the address of the resource (document) the user wants and it needs a way to communicate with the server. The address is known as the Uniform Resource Locator (URL) and the means of communication is through a protocol such as HTTP. The URL is actually present in the document being displayed. When a user clicks on a hypertext link, the browser gets the URL from the document. Given the protocol and address, the browser transmits a request to open a connection to the server. Once the connection is made, the browser sends a request for the document. The server sends the document, if it exists, and disconnects from its end of

the connection. As previously stated the first information that you get from a remote Web server is known as a home page. The home page is an initial interface to a series of other documents, files, and resources that reside on that computer or on other Web servers around the world.

The document which is returned by the server is formatted with HTML (Hypertext Markup Language) commands. HTML is a computer programming language used to create, format, and identify titles, subheadings, bold, italics, and hyperlinks that enable you to jump between places within a document, on the same computer (server), or to another remote server located somewhere on the Internet. HTML is comprised of a standard set of codes or "tags" that are inserted into a document and leaves the interpretation of these marked elements up to the browser displaying the document. World Wide Web pages are written and composed in the HTML page description language. Despite the fact that different browsers understand more or less sophisticated versions of the HTML language, the de facto standard for Web browsers is that they interpret HTML documents at the very least.

Unfortunately, not every Web browser works in the same way. Some browsers support graphics, other do not. Some browsers have extensive formatting support features while other have just a few. Some do not support the more advanced features.

HTML compliant Web browsers parse the stream of commands and use tools from their native operating environment to build display images representing the material in the command stream. Furthermore, HTML compliant Web Browsers typically expect a webpage, formatted in HTML, to come either from one of the many servers on the Internet network, or from a file. In either situation, the HTML stream must define where the page is coming from. If the webpage can come from a variety of media (CD-ROM,DVD) a regular browser and standard HTML could not handle that.

Another shortcoming of present browser technology concerns the finite time period between the time a user initiates a link and the return of the webpage. We live in a world where speed and convenience has become the norm. As the economies of nations around the world become increasingly interdependent, speed of information delivery becomes a valuable competitive tool. While the Internet has no peer with regard to moving massive amounts of information rapidly, most users of the Internet still experience varying degrees of frustration over the delays inherent in the time required to transfer documents across the network. The amount of time it takes to transfer an image across a network depends on a number of factors, amongst them are the transfer rate, the quantity of data, network traffic and the throughput speed of storage devices. Since networks are usually comprised of many sub-networks, it is extremely

difficult to estimate actual data transfer rates. It may be that the backbone of the network is fiber optic, but if the link from the backbone to the host or client computer is something slower like ethernet or much worse a modem, rates will suffer.

The second most influential factor affecting data transfer rates is the quantity of data transferred. The more data you have, the longer it takes to transfer it across a network. Digital images, for example, are extremely data intensive. Some Web documents have so many inline images, or a very few large ones, that they can take several minutes to load on a slow (modem) connection. While data reduction techniques may be employed to lessen the impact the tradeoff is image quality.

A third variable, over which the user has no control concerns network traffic. The combination of the above factors create a situation which impedes the exchange of information between clients and servers on the World Wide Web and frustrates end users as a result.

Another shortcoming associated with existing browser technology concerns the limited recognition of the ever growing list of protocols with which multimedia resources are transported and displayed over the internet. The World Wide Web is generally considered to consist primarily of the following three services, HTTP, Gopher, and FTP. Popular Browsers, like Netscape Navigator TM utilize the HTTP protocol to access hypertext documents from Web servers. HTTP is the service that enables Web clients to receive hypertext content that can link to other hypertext content and to nonhypertext content as well. HTTP is just an Internet protocol like so many other protocols. It's popularity stems mainly from its early use. There exists today a number of more efficient protocols not currently supported by the current generation of browsers. It would be desirable for a regular browser to be able to access resources that utilize new or existing protocols other than HTTP.

A further shortcoming associated with the internet is that of security. Security, and its many implications, has become and remains a real concern since the very inception of the Internet. Parents are concerned that their children run the risk of gaining easy access, or being inadvertently exposed to, ever increasing amounts of adult oriented information not intended for minors. Employers' concern run to their employees who must be constrained from utilizing the internet for purposes other than to satisfy their employer's needs. Given the widespread corporate acceptance of the Internet, unproductive use of the resource at the corporate level could translate into billions of dollars of lost revenue.

In addition to the concerns of multiple protocols, security, transfer speed, and non-HTML datastreams, a further concern relates to the issue of file compression. While compressing and decompressing files are a necessary consequence of today's file sizes and transmission rates. It would be desirable for a user to not have to be concerned with the vagaries of the different compression formats and versions in existence today.

Accordingly the invention provides a system for supporting an HTML web browser and providing HTML formatted content thereto from a variety of data resources as requested by a user, said system running in a layer between said browser and an operating system for a personal computer, said system comprising: (a) a search engine for: (i) receiving data requests comprising web pages with associated URL addresses from said HTML web browser; (ii) locating data from said variety of data resources requested by said HTML browser; (iii) retrieving data requested by said HTML browser from said variety of data resources; (iv) receiving data broadcast from said variety of data resources; (b) means for translating said retrieved data to HTML format when HTML formatting is not found; (c) means for performing a decompression on retrieved data when said data is compressed; (d) means for providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table; (e) means for translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found; (f) means for transparently retrieving web pages whose URL address matches the URL associated with said data requtest from the HTML web browser; and (g) means for amending said data requested and broadcasting from said variety of resources.

In a further aspect, the invention provides a method for supporting an HTML web browser and providing HTML formatted content thereto from a variety of data resources as requested by a user, the method comprising the steps of: (a) receiving data requests comprising web pages with associated URL addresses from said HTML web browser; (b) locating data from said variety of data resources requested by said HTML browser; (c) retrieving data requested by said HTML browser from said variety of data resources; (d) receiving data broadcast from said variety of data resources; (e) translating said retrieved data to HTML format when HTML formatting is not found; (f) performing a decompression on retrieved data when said data is compressed; (g) providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table; (h) translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found; (i) transparently retrieving web pages whose URL address matches the URL associated with said data request from the HTML web browser; and (j)

amending said data requested and broadcast from said variety of resources.

In a yet further aspect, the invention provides a computer program device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for supporting an HTML web browser and providing HTML formatted content thereto from a variety of data resources as requested by a user, said program of instructions running in a layer between said browser and an operating system for a personal computer, the method comprising the steps of: (a) receiving data requests comprising web pages with associated URL addresses from said HTML web browser; (b) locating data from said variety of data resources requested by said HTML browser; (c) retrieving data requested by said HTML browser from said variety of data resources; (d) receiving data broadcast from said variety of data resources; (e) translating said retrieved data to HTML format when HTML formatting is not found; (f) performing a decompression on retrieved data when said data is compressed; (g) providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table; (h) translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found; (i) transparently retrieving web pages whose URL address matches the URL associated with said data request from the HTML web browser; and (j) amending said data requested and broadcast from said variety of resources.

A preferred embodiment of the present invention enhances the operation of a web browser by providing means for recognizing URL requests from diverse media sources including a hard disk cache, and read only media including CD-Rom and DVD from a network.

The operation of a web browser can be further enhanced by providing means for modifying an incoming data stream (e.g. to allow advertisements into a stream, or add a set of consistent "buttons" to all HTML pages, or add JAVA applications including smart printing, site viewers, and smart site searching). The operation of a web browser can be still futher enhanced by providing a capability for decrypting downloaded information in encrypted format to allow access to compressed media content like CD Roms and DVD.

A preferred embodiment of the present invention enhances the operation of a web browser by providing a security mechanism which allows a user access to only specific media or portions of media; by providing means for accessing information utilizing protocols other than the Hypertext Transfer Protocol (HTTP); and by providing means for enhancing a users perception of download speed associated with user requested HTML pages.

The approach of the preferred embodiment is to use a Micro Web Server (hereinafter MWS) that runs on the same machine as the browser, but in a separate process. This allows for local processing to occur before the HTML data stream reaches the browser, thereby allowing for manipulation of that string, while still using standard off the shelf browser technology. The MWS effectively sits between the browser and the data that the browser requests. From the browser's perspective all data is coming from a web server when in fact the data may be coming from a set of diverse sources. The MWS also allows for "intelligent" brokering between a browser and a datastream.

The MWS also allows for translation of different protocols to HTTP and the different data types to HTML, which would allow access and display of documents that couldn't normally be displayed using a standard HTML browser (e.g. IBM 3270 screen format and protocol).

This mechanism also allows for a hybrid remote-local media paradigm, where a backend (e.g. a DB2 database) can reside wholly on local media, or be split between local and remote media. Furthermore, this mechanism can allow for translation of URL's before they are actually fetched from the Web.

The MWS mechanism can further enhance the perception of page download time by intelligently pre-fetching links. That is, whenever a page is fetched by the MWS, any reference to a URL is noted. After the page is sent to the browser for display, the MWS goes through the list of possible links and starts pre-fetching those links and caching them on the local drive.

The MWS can also be used to enhance the printing of a Web document. If a user requests that a whole document be printed, the MWS can use a similar analysis as in the "Intelligent pre-fetch of Links" to fetch the whole document and print it in the background.

A preferred embodiment of the invention is implemented in a computer as a computer program stored in a computer readable format where the computer program consists of program statements executable by the computer. The computer preferably comprises a processor, an operating system, multiple local storage devices, a graphical user interface, a means for remote connection to the Internet, and an HTTP-compliant browser.

A preferred embodiment of the present invention will now be described in detail by way of example only with reference to the following drawings:

FIG 1.a is a schematic prior art diagram of a communication model which describes the Internet.

FIG 1.b is a schematic diagram of a communication model which describes the Internet incorporating a preferred embodiment of the present invention, including a micro web server.

FIG 2. illustrates a representative client computer supporting an HTML compliant Worldwide Web browser.

FIG 3. illustrates the functional enhancements provided by the micro web server, where each functional enhancement is represented by a dedicated kernel.

FIG 4. is a flowchart diagram of the operation of a preferred embodiment of the present invention for information incoming to a user, via a micro-web server, from one of a multiplicity of sources.

FIG 5. is a flowchart of a preferred embodiment of the present invention for processing outgoing information from a user, via a microweb server, MWS, to one of a multiplicity of sources.

FIG. 1 is a block diagram illustrating a representative "client" computer 1 in which a preferred embodiment of the present invention is implemented. The system unit 1 includes a system bus 2 to which various components are coupled and by which intercommunication between the devices occurs. The RAM 3 is the main memory into which both the operating system 4 and application programs such as the browser 5 and Micro Web Server 6 are loaded. The RAM may also serve the additional function of storing (caching) downloaded information from the Internet as requested by the user. Also shown in FIG. 2 is the hard disk drive 7, floppy disk 8, CD-ROM 9, and DVD 10 as representative of typical local storage mediums.

The operating system 4 of the "client" computer may be any known or available operating system. The browser in a preferred embodiment is considered to be HTML and HTTP compliant. Known browser software includes Netscape, Netscape Navigator 2.0 and 3.0, Internet Explorer, and Mosaic to name some of the more popular browsers currently in use. A preferred embodiment of the present invention is designed to operate in concert with any of these known or developing web browsers.

According to the preferred embodiment, a micro web server 6, runs on the client computer 1 along with the browser 5 but in a separate process. This allows for local processing of the incoming data stream to occur before the HTML data stream reaches the browser 5, thereby allowing for manipulation of that string, while still using standard off the shelf browser technology.

In the preferred embodiment, a user is the option of selecting whether or not the features of the present invention are to be utilized in conjunction with the standard browser, in essence, whether the present invention is to be "switched on". In the preferred embodiment, certain

options are presented with regard to password protection checking. The first operation the MWS performs when trying to fetch a page is to check if the user is allowed to access the page. A user can be given a password protected list of pages and sites that are allowed to be viewed or lists of pages and sites that are not allowed to be viewed. This could also encompass a list given to the user after a fee is paid, so that certain sites have a viewing payment associated with them. When a page is fetched it is checked against the list. If the page is not viewable a page describing the problem is sent to the browser. The MWS could also keep a list of all the pages requested for later review and monitoring.

Fig. 2a is a prior art illustration of the communication model which best describes the Internet, that of a client/server architecture where a network of browsers and servers work together for the purpose of transmitting and receiving information. Fig. 2a illustrates a typical client/server pair where a browser 14 running as a client program on a client computer 12, requests information, via the HTTP protocol 16, from a server program 20 running on another computer 22 somewhere on the Internet. That server sends the requested data back over the Net, via an HTML style document utilizing the HTTP protocol 18, to the Web browser, which interprets and displays the data on the client's screen. From the client's perspective browsing the Web entails two basic hardware requirements: it must have a connection to the Internet and must be capable of running a Web browser program.

Fig. 2b is an illustration of a communication model of the Internet incorporating the apparatus of a preferred embodiment of the present invention. The client/server architecture depicted in Fig. 2b is similar to that described in Fig. 2a, however, Fig. 2b illustrates the inclusion of a micro web server, MWS, 31 running as a client program on the same client computer 12, but in a separate process, along with the standard browser 14. Browser initiated requests 16 are first intercepted by the MWS 31 for any required pre-processing prior to the request's issuance to a diversity or destinations including a remote web server 38 running on a remote computer 40, other media 42 (including CD-Rom, DVD), a file system 44 internal to the client computer 24 initiating the request, or other servers in the network 46. As a corollary to the HTTP requests which originate from the browser 26, HTML replays to be received by the Browser 26 are first routed from one of the diversity of sources described in Fig. 2b as an HTML or non HTML data stream, depending upon the source to the MWS 31 for any required pre-processing prior to the MWS passing the data stream as an HTML reply 30 to the standard Browser 26.

Fig. 3 illustrates the functional enhancements provided by the incorporation of the MWS 52 as an intermediate process between the standard browser 50 and the various external and internal media (e.g. 66,

68, 70, and 72 described at Fig. 2). According to a preferred embodiment of the present invention, whenever a standard browser requests a new page one or more functions may be performed by various kernels contained within the MWS 52 to facilitate the request. Those functions (kernels) include:

1) Kernel 1 - URL lookup 54 - The MWS accepts URL requests and in turn decides what media the source is coming from (e.g. harddisk cache HD, read only media, such as CD-Rom or DVD, or from the network, thereby allowing a mixed media search (e.g. first HD, then CD or DVD, then the Web). Once the source has been determined the appropriate page is transmitted via the MWS to the Browser. This functionality facilitates the reception of Web pages broadcast via mediums including satellite or cable.

2) Kernel 2 - Data Stream modification 56 - for data streams received by the MWS from external sources, the MWS prior to passing the data stream on to the standard browser modifies the data stream in a number of ways. Those modifications are not exclusive to but could include adding advertisements into the stream, adding a set of consistent buttons to all or a pre-specified number of HTML pages, and/or adding JAVA apps (e.g. smart printing, site viewers, smart site searching), 2)

3) Kernel 3 - Site/Page Background Fetch 58 - a users perception of network speed is enhanced by analyzing the page a user is currently accessing and prefetching links associated with that page in the background. This prefetching can be preempted at any time by a user request to fetch a specific page.

4) Kernel 4 - Compression 60 - whenever a datastream is received from media other than a web server (e.g. local hard drive, CD-Rom, DVD) that datastream may be compressed. The MWS performs any necessary decompression of the datastream as the datastream is being received.

5) Kernel 5 - Security 62 - where access by a client to only specific data sources or types is allowed. The first operation performed by the MWS when trying to fetch a page requested by the browser is to check if the user is allowed to access the page. A user can pre-store in the MWS a password protected list of pages and sites that are allowed to be viewed or alternatively a list of pages and sites that are not allowed to be viewed. This could also be a list of allowable sites accessible to a user after payment of a site fee for those sites which have a viewing payment associated with them. When a page is requested by the browser 2 it is first checked against the list. If the page is not deemed viewable a page describing that situation is sent to the browser. The MWS could additionally keep a list of all the pages requested for later review and monitoring.

6) Kernel 6 - Translation 64 - one or more automatic translators are incorporated into the MWS so that web pages can be translated as they are obtained from the web. Depending on the speed of the translation

engine, the translation could either be done on the fly, or when the page is cached on the local disk.

Each of the aforementioned enhancements provided by the MWS will be described in greater detail in the illustrations described herein.

Referring to FIG. 4, a flowchart illustrates a preferred operation of the MWS depicting the operations performed for a data stream arriving to a user from some external source. At step 73 the MWS receives an incoming data stream. At step 74, the MWS first analyzes the incoming data stream to determine whether the data stream is compressed. If so, the operation continues at step 76 where the incoming data stream is decompressed utilizing an appropriate decompression algorithm which is stored internally in the MWS. In the preferred embodiment a plurality of decompression algorithms are pre-stored in the MWS. If the incoming data stream is not compressed operation continues at decision step 78 where a determination is made as to whether the incoming data stream is in HTML format. If so, operation continues at step 82, otherwise, a translation would occur as depicted at step 80. The MWS would typically incorporate translators for a number of standard screen and terminal protocols including 3270, 5250, and VT100.

If the incoming data stream is already in HTML format operation continues at step 82 where a determination is made as to whether there is any secondary HTML source code. The additional code to be added to the primary incoming datastream could comprise additional HTML commands, Java source code, or Java Script source code. This secondary stream of source code to be merged could be stored in the servers file system, or on a DVD, or as an additional page fetched from the server. The secondary code could enhance the primary datastream in a number of ways including : advertisements, standard buttons such as "update page", "Update site", "Connect to Page", or possibly a site map or a smart print button. The MWS could contain a ruleset for determining when to insert or not insert a secondary codestream. In an illustrative embodiment the ruleset could decide to always add the secondary stream, or it could otherwise decide to only add the secondary codestream for certain URL's. The ruleset is not limited to these alternatives, it could feasibly choose any user specified ruleset.

If secondary source code needs to be added to the primary datastream operation continues at step 84 where the primary data stream is amended, otherwise operation continues at step 86 where the data stream is then passed on to the resident Browser for further processing.
Having established that the requested data is not resident on any of the local storage mediums the webpage must then be downloaded from a remote server. In addition to this process, the MWS performs a "pre-fetch"

operation in a background (transparent) mode 81. The pre-fetch operation requires that the MWS download the data associated with each hypertext link whose URL matches the URL of the explicit user website request. The downloaded information is cached locally in anticipation of a potential client request at some future point in the current internet session. The "pre-fetch" algorithm effectively streamlines the client/server session by making information instantly available if and when requested by the client. The algorithm eliminates the "downtime" associated with file transfer across the internet.

The "pre-fetch" process is considered to be user-transparent in that the process occurs independent of any explicit user initiated requests.

The "pre-fetch" algorithm restricts downloaded information to include those links associated with the currently accessed website. Associated is defined by the URL's having the same host address where the home address is located in the URL between the first set of double forward slashes and the next single forward slash (/). The home address refers to the specific host computer where the requested data resides. The remaining portion of the address details the path (directory) on the computer to the specific resource desired. Any non-associated hypertext links would be ignored by the pre-fetch algorithm. In those situations where downloading the total information content associated with a particular website would prove impractical given the memory limitations of most computers the information downloaded to the user would be restricted only to the extent that sufficient space is available on the local storage medium. Some user defined upper bound may be established beforehand.

As previously stated, the "prefetch routine" of the MWS described above restricts its selection of links to only those hypertext links containing associated pages. In an alternative preferred embodiment however, this restriction may be modified to download data defined by non-associated links based on some pre-established criteria (e.g. statistical test, most recently used algorithm). For example, the selection algorithm could employ a statistical test to decide which hypertext links to select or reject. The statistical test would access some previously defined usage table built into the MWS to decide the historical frequency of access of particular links and select only those links which exceed some user defined threshold.

Another test, for example, might select hypertext links based solely on those links most recently accessed. Where "most recent" would also be user definable. Other tests may be constructed on the basis of user preference or any other user defined criteria.

Referring to FIG. 5, a flowchart illustrates a preferred embodiment of the operational steps performed by the MWS for outgoing data. The flowchart of Fig. 5 shows the process by which user requests are routed to some external server in accordance with the method of the present invention. A client computer typically initiates a request for information from one of a plurality of servers on the World Wide Web.

A request is issued by a client via a standard browser at step 88. At step 90 the request is first routed to the MWS. Once the request is received by the MWS the process continues to step 92 where a determination is made as to whether the URL request contains a request for an MWS function. That is, for any functionality that may have been incorporated into the standard web browser by the MWS, this step checks whether the user attempted to exercise the MWS supplied functionality. For example, the MWS may have inserted a print button, the MWS has to be able to act on a print request when issued by a client. The MWS could be embedded as part of the returned URL address, and then stripped off for execution at step 94. Otherwise operation continues at decision step 96, where a determination is made as to whether a local (cached) copy of the requested data exists on one of the local storage mediums (hard-drive, CD-Rom, DVD) by searching an internal URL lookup table. If a local copy exists the MWS must first, according to the preferred embodiment, do a local copy lookup using the parameters of the internal URL lookup table at step 98. The operation continues at decision step 100 where a determination is made as to whether the network (downloaded) version contains a version date. If so, the operation continues at step 104 where the MWS compares the version date of the local copy with the network copy to decide whether the network version date is earlier. If so, operation continues at step 108 where the network version is selected for both storage and display. Otherwise, the local version is selected at step 106, for storage and display. If there is no version date associated with the local lookup the process continues at decision step 102 where a determination is made as to whether the network page has been "pre-fetched". If so, operation continues at step 110 where the pre-fetched network page is stored and displayed. Otherwise operation continues at step 112 where the MWS will select the network page for storage and display. The process then terminates at step 120. The only situation that involves displaying the local copy is one where the user is not currently connected to the network.

In the situation where no local copy exists, operation continues at decision step 114 where the MWS performs a user authorization check. The webpage requested is checked against a table, internal to the MWS, consisting of restricted, non-authorized webpages for that user. If the current user request is contained on this list the MWS returns a message to the user indicating that the webpage is not authorized for viewing for

that particular user at step 115. The process then terminates at step 120. Otherwise, the requested webpage is considered authorized and therefore viewable by the user. In the event that the request is considered authorized operation continues at decision step 116 where a determination is made as to whether the requested page is password protected. If so, operation continues at step 118 where the user's ID and password are then checked by the MWS against an internal table to establish the identity of the client. At step 124 a determination is made as to the validity of the user's ID and password. If the user supplied ID and password prove invalid operation continues at step 122 where an error message is displayed. The process then terminates at step 120. Otherwise, given that the user has supplied a valid ID and password, operation continues at step 126 where a determination is made as to whether the outgoing datastream requires a translation. The translation is required in those instances where the protocol of the incoming datastream is something other than HTML. If a translation is not required operation continues at step 130 where the request is forwarded to the server to be fulfilled. Otherwise an appropriate translation will be performed at step 128 on the outgoing datastream. After any required translations are performed operation continues at step 130 where the request is forwarded to the remote server. The process then terminates at step 120.

CLAIMS

1. A system (31) for supporting an HTML web browser (26) and providing HTML formatted content thereto from a variety of data resources (40, 42, 44, 46) as requested by a user, said system running in a layer between said browser and an operating system for a personal computer, said system comprising:

(a) a search engine (53) for:

(i) receiving data requests comprising web pages with associated URL addresses from said HTML web browser;

(ii) locating data from said variety of data resources requested by said HTML browser;

(iii) retrieving data requested by said HTML browser from said variety of data resources;

(iv) receiving data broadcast from said variety of data resources;

(b) means (54) for translating said retrieved data to HTML format when HTML formatting is not found;

(c) means (56) for performing a decompression on retrieved data when said data is compressed;

(d) means (58) for providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table;

(e) means (60) for translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found;

(f) means (62) for transparently retrieving web pages whose URL address matches the URL associated with said data request from the HTML web browser; and

(g) means (64) for amending said data requested and broadcasting from said variety of resources.

2. The system of claim 1, wherein said data resources include at least one local data storage device and at least one remote server.

3. The system of claim 2, wherein said local data storage device includes one of a hard disk, a DVD, or a CD-Rom.

4. The system of claim 2 or 3, further comprising means for utilizing a local version of said requested data on said local storage device when one exists.

5. The system of claim 4, where the means for utilizing a local version of said requested data when one exists on said local storage device includes:

(i) means for accessing a lookup table internal to said system local storage device to determine if a local version exists;

(ii) means responsive to the existence of a local version for determining whether there is an associated version date; and

(iii) means responsive to an associated version date for comparing the associated version date with the version date associated with a downloaded web page from said remote server.

6. The system of claim 5, wherein said web page associated with the more current version date is stored and displayed.

7. The system of any preceding claim, wherein said user defined access table contained in said providing means comprises a list of unauthorized URL web pages.

8. The system of claim 7, wherein said providing means denies said user request when said request is made for one of said unauthorized URL web pages.

9. The system of any of claims 1 to 6, wherein said user defined access table contained in said providing means comprises a list of authorized URL web pages.

10. The system of claim 9, wherein said providing means executes said user request when said request is made for one of said authorized URL web pages.

11. The system of any preceding claim, wherein said retrieving means further comprises an algorithm for determining which URL locations on a downloaded webpage will be searched by said search engine.

12. The system of any preceding claim, wherein said performing means further comprises one or more algorithms for decompressing HTML data associated with a web page from a remote server.

13. The system of claim 12, wherein said algorithms include statistical tests or a most recently accessed test.

14. The system of any preceding claim, wherein said amending means amends said data requested and broadcast from said variety of resources with secondary HTML source code comprising HTML commands, JAVA source code.

15. A method for supporting an HTML web browser (26) and providing HTML formatted content thereto from a variety of data resources as requested by a user, the method comprising the steps of:

(a) receiving data requests comprising web pages with associated URL addresses from said HTML web browser; (73)

(b) locating data from said variety of data resources requested by said HTML browser;

(c) retrieving data requested by said HTML browser from said variety of data resources;

(d) receiving data broadcast from said variety of data resources;

(e) translating said retrieved data to HTML format when HTML formatting is not found;

(f) performing a decompression on retrieved data when said data is compressed; (76)

(g) providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table;

(h) translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found; (80)

(i) transparently retrieving web pages whose URL address matches the URL associated with said data request from the HTML web browser;(81) and

(j) amending said data requested and broadcast from said variety of resources (84).

16. The method of claim 15, wherein said data resources include at least one local data storage device and at least one remote server.

17. The method of claim 16, wherein said local data storage device includes one of a hard disk, a DVD, or a CD-Rom.

18. The method of claim 16 or 17, further comprising means for utilizing a local version of said requested data on said local storage device when one exists.

19. The method of claim 18, where the means for utilizing a local version of said requested data when one exists on said local storage device includes:

(i) accessing a lookup table internal to the local storage device to determine if a local version exists; (98)

(ii) if a local version exists determining whether there is an associated version date; (100) and

(iii) if there is an associated version date comparing the associated version date with the version date associated with a downloaded web page from said remote server (104).

20. The method of claim 19, wherein said web page associated with the more current version date is stored and displayed.

21. The method of any of claims 15 to 20, wherein said user defined access comprises a list of unauthorized URL web pages.

22. The method of claim 21, wherein said user request is denied when said request is made for one of said unauthorized URL web pages.

23. The method of any of claims 15 to 20, wherein said user defined access table comprises a list of authorized URL web pages.

24. The method of claim 23, wherein said user request is executed when said request is made for one of said authorized URL web pages.

25. The method of any of claims 15 to 24, wherein the transparently receiving step further comprises an algorithm for determining which URL locations on a downloaded webpage will be searched by said search engine.

26. The method of any of claims 15 to 25, wherein said decompression step further comprises one or more algorithms for decompressing HTML data associated with a web page from a remote server.

27. The method of claim 26, wherein said algorithms include statistical tests or a most recently accessed test.

28. The method of any of claims 15 to 27, wherein the step of amending said data requested and broadcast from said variety of resources further comprises amending said data with secondary HTML source code comprising HTML commands, JAVA source code.

29. A computer program device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for supporting an HTML web browser and providing HTML formatted content thereto from a variety of data resources as requested by a user, said program of instructions running in a layer between said browser and an operating system for a personal computer, the method comprising the steps of:

(a) receiving data requests comprising web pages with associated URL addresses from said HTML web browser; (73)

(b) locating data from said variety of data resources requested by said HTML browser;

(c) retrieving data requested by said HTML browser from said variety of data resources;

(d) receiving data broadcast from said variety of data resources;

(e) translating said retrieved data to HTML format when HTML formatting is not found;

(f) performing a decompression on retrieved data when said data is compressed; (76)

(g) providing selective access to specific web pages or portions of specific web pages as determined by a user defined access table;

(h) translating one of a plurality of internet protocols to the HTTP protocol if HTTP protocol is not found; (80)

(i) transparently retrieving web pages whose URL address matches the URL associated with said data request from the HTML web browser; (81) and

(j) amending said data requested and broadcast from said variety of resources (84).

30. The computer program device of claim 29, wherein said data resources include at least one local data storage device and at least one remote server.

31. The computer program device of to claim 30, wherein said local data storage device includes one of a hard disk, a DVD, or a CD-Rom.

32. The computer program device of claim 30 or 31, further comprising means for utilizing a local version of said requested data on said local storage device when one exists.

33. The computer program device of claim 32, where the means for utilizing a local version of said requested data when one exists on said local storage device includes:

(i) accessing a lookup table internal to said local storage device to determine if a local version exists;

(ii) if a local version exists determining whether there is an associated version date; and

(iii) if there is an associated version date comparing the associated version date with the version date associated with a downloaded web page from said remote server.

34. The computer program device of claim 33, wherein said web page associated with the more current version date is stored and displayed.

35. The computer program device of any of claims 29 to 34, wherein said user defined access comprises a list of unauthorized URL web pages.

36. The computer program device of claim 35, said user request is denied when said request is made for one of said unauthorized URL web pages.

37. The computer program device of any of claims 29 to 34, wherein said user defined access table comprises a list of authorized URL web pages.

38. The computer program device of claim 37, wherein said user request is executed when said request is made for one of said authorized URL web pages.

39. The computer program device of any of claims 29 to 38, wherein the transparently receiving step further comprises an algorithm for determining which URL locations on a downloaded webpage will be searched by said search engine.

40. The computer program device of any of claims 29 to 39, wherein said decompression step further comprises one or more algorithms for decompressing HTML data associated with a web page from a remote server.

41. The computer program device of claim 40, wherein said algorithms include statistical tests or a most recently accessed test.

42. The computer program device of any of claims 29 to 41, wherein the step of amending said data requested and broadcast from said variety of resources further comprises amending said data with secondary HTML source code comprising HTML commands, JAVA source code.

The Patent Office

INVESTOR IN PEOPLE

20

Application No: GB 9912801.9
Claims searched: All

Examiner: Gareth Griffiths
Date of search: 9 November 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): H4P (PPEC, PPG)

Int Cl (Ed.6): G06F 17/30, H04L 12/56, 29/06

Other: Online Databases: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A,P	EP0921481 A (IBM)	
A	WO97/18635 A (IBM)	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.